High-order linear and non-linear Residual Distribution schemes for the simulation of compressible viscous flows

Dante De Santis

Stanford University Aeronautics and Astronautics

AMS Seminar
March 26, 2015

Background & motivation
CFD as a common tool in engineering

Today industrial practice

- FV schemes (theoretical) 2nd order accuracy
- Not accurate enough for realistic applications
- Very fine meshes required: CPU expensive calculations
- Simulations of large problems still expensive for fast design & optimization

Background & motivation
CFD as a common tool in engineering

Today industrial practice

- FV schemes (theoretical) 2nd order accuracy
- Not accurate enough for realistic applications
- Very fine meshes required: CPU expensive calculations
- Simulations of large problems still expensive for fast design & optimization

Appeal of high-order methods

- Faster reduction of the discretization error with the DoFs
- Accurate solutions at acceptable costs
- High-fidelity simulations make possible to check deficits of physical modeling

Background & motivation
HO methods for real-life applications

- Several high-order schemes developed
  - ENO/WENO
  - Continuous finite elements
  - Discontinuous Galerkin
  - Spectral volume, Spectral difference, Flux reconstruction, . . .

Background & motivation
HO methods for real-life applications

- Several high-order schemes developed
  - ENO/WENO
  - Continuous finite elements
  - Discontinuous Galerkin
  - Spectral volume, Spectral difference, Flux reconstruction, ...

Active research field

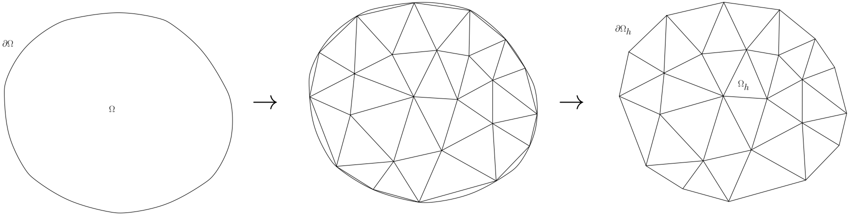
- Reduce the computational costs
- Reliable discretization of discontinuous solutions
- RD: A possible solution to these limitations

Presentation outline

Residual distribution schemes

The main idea for steady problems

- Solve $\nabla \cdot \boldsymbol{f}(u) = 0$, on $\Omega \subset \mathbb{R}^d$



Residual distribution schemes
The main idea for steady problems

- Solve $\nabla \cdot \boldsymbol{f}(u) = 0$, on $\Omega \subset \mathbb{R}^d$
- Solution approximated on each element with piece-wise continuous polynomials of k-th order: $u \simeq u_h(\boldsymbol{x})$

$$u_h = \sum_{i \in \mathcal{N}_h} \psi_i(\boldsymbol{x})\, u_i,$$

Residual distribution schemes

The main idea for steady problems

- Solve $\boldsymbol{\nabla} \cdot \boldsymbol{f}(u) = 0$, on $\Omega \subset \mathbb{R}^d$
- Solution approximated on each element with piece-wise continuous polynomials of k-th order: $u \simeq u_h(\boldsymbol{x})$
- The approximated solution will give birth to a residual: $\Phi^e(u_h)$

$$\Phi^e(u_h) = \int_{\Omega_e} \boldsymbol{\nabla} \cdot \boldsymbol{f}(u_h)\, \mathrm{d}\Omega = \int_{\partial\Omega_e} \boldsymbol{f}(u_h) \cdot \hat{\boldsymbol{n}}\, \mathrm{d}\Omega$$

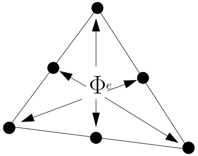
Residual distribution schemes

The main idea for steady problems

- Solve $\nabla \cdot \boldsymbol{f}(u) = 0$, on $\Omega \subset \mathbb{R}^d$
- Solution approximated on each element with piece-wise continuous polynomials of k-th order: $u \simeq u_h(\boldsymbol{x})$
- The approximated solution will give birth to a residual: $\Phi^e(u_h)$
- Distribution to the DoFs of the element residual

$$\Phi_i^e = \beta_i^e \Phi^e$$

$$\sum_{i \in \mathcal{N}_h^e} \Phi_i^e = \Phi^e, \text{ for conservation}$$

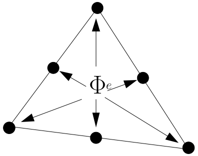


Residual distribution schemes
The main idea for steady problems

- Solve $\boldsymbol{\nabla} \cdot \boldsymbol{f}(u) = 0$, on $\Omega \subset \mathbb{R}^d$
- Solution approximated on each element with piece-wise continuous polynomials of k-th order: $u \simeq u_h(\boldsymbol{x})$
- The approximated solution will give birth to a residual: $\Phi^e(u_h)$
- Distribution to the DoFs of the element residual

$$\Phi_i^e = \beta_i^e \Phi^e$$

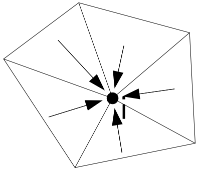
$$\sum_{i \in \mathcal{N}_h^e} \Phi_i^e = \Phi^e, \text{ for conservation}$$



- Gather the residuals: $\displaystyle\sum_{i \in \mathcal{N}_h^i} \Phi_i^e = 0, \quad \forall i$

Change of the solution driven by non-zero element residual

$$\frac{u_i^{n+1} - u_i^n}{\Delta t^n} + \sum_{i \in \mathcal{N}_h^i} \Phi_i^e = 0, \;\; n \to \infty$$



Residual Distributions schemes

Some approach for advection-diffusion problems

- Well established methodology for advection problems
- What about the discretization of advection-diffusion problems?

$$\boldsymbol{\nabla} \cdot \boldsymbol{f}(u) = \boldsymbol{\nabla} \cdot \big(\nu \boldsymbol{\nabla} u\big), \qquad \mathrm{Pe} = \|\boldsymbol{a}\|\, h/\nu$$

Residual Distributions schemes
Some approach for advection-diffusion problems

- Well established methodology for advection problems
- What about the discretization of advection-diffusion problems?

$$\boldsymbol{\nabla} \cdot \boldsymbol{f}(u) = \boldsymbol{\nabla} \cdot \left(\nu \boldsymbol{\nabla} u\right), \qquad \mathrm{Pe} = \|\boldsymbol{a}\|\, h/\nu$$

- Old approach: mixed RD (for advection) and Galerkin (for diffusion)
  - Error analysis reveals that the approach is 1st order accurate when $\mathrm{Pe} \sim 1$
  - Proper scaling of the RD upwind stabilization with Pe (Ricchiuto et al.)
- New principle [Roe, Nishikawa, Caraeni, ...] one distribution process for the residual of whole equation (advection+diffusion) to get an uniform order of accuracy on entire spectrum of Pe

Discretization of Advection-Diffusion problems

Calculation of the total residual

- Total residual of the advection-diffusion problem

$$\Phi^e = \int_{\Omega_e} \left(\boldsymbol{\nabla} \cdot \boldsymbol{f}\left(u_h\right) - \boldsymbol{\nabla} \cdot \left(\nu \boldsymbol{\nabla} u_h\right)\right) \mathrm{d}\Omega.$$

Discretization of Advection-Diffusion problems

Calculation of the total residual

- Total residual of the advection-diffusion problem

$$\Phi^e = \int_{\Omega_e} \left(\boldsymbol{\nabla}\cdot \boldsymbol{f}(u_h) - \boldsymbol{\nabla}\cdot\left(\nu \boldsymbol{\nabla} u_h\right)\right) \mathrm{d}\Omega.$$

- Using the divergence theorem

$$\Phi^e = \oint_{\partial\Omega_e} \left(\boldsymbol{f}(u_h) - \nu \widetilde{\boldsymbol{\nabla} u_h}\right)\cdot \hat{\boldsymbol{n}}\, \mathrm{d}\partial\Omega,$$

- For a piece-wise polynomial interpolation, $\boldsymbol{\nabla} u^h \cdot \boldsymbol{n}$ is discontinuous at the elements face. The scheme requires a continuous gradient $\widetilde{\boldsymbol{\nabla} u^h}$

RD: Distributions process

Central linear schemes (inspired by Lax-Wendroff)

- Originally proposed for multidimensional upwinding
  - Roe, Deconinck, Abgrall ...
  - Formulation on simplexes & difficult extension to HO
- Central schemes: HO & general elements

RD: Distributions process

Central linear schemes (inspired by Lax-Wendroff)

- Originally proposed for multidimensional upwinding
  - Roe, Deconinck, Abgrall ...
  - Formulation on simplexes & difficult extension to HO
- Central schemes: HO & general elements

Linear-scheme

$$\Phi_i^{e,\mathrm{LW}} = \frac{\Phi^e}{N^e_{\mathrm{dof}}} + \int_{\Omega_e} \boldsymbol{a}\cdot\boldsymbol{\nabla}\psi_i\,\tau\left(\boldsymbol{a}\cdot\boldsymbol{\nabla}u_h - \boldsymbol{\nabla}\cdot\left(\nu\widetilde{\boldsymbol{\nabla}u^h}\right)\right)\mathrm{d}\Omega$$

$$\tau = \frac{1}{2}\frac{|\Omega_e|}{\sum\limits_{j\in\mathcal{N}_h^e}\max(k_j,0)+\nu}, \quad \text{with} \quad k_j = \frac{1}{2}\,\bar{\boldsymbol{a}}\cdot\boldsymbol{n}_j,$$

- Reconstructed gradient used in the stabilization term
- High-order preserving: $u_h \in \mathbb{P}^k \Rightarrow$ scheme $\mathcal{O}(h^{k+1})$ (always?)
- Linear scheme: not monotone on shocks

RD: Distributions process

Construction of a non-linear RD scheme

- First order monotone scheme, *e.g.* Rusanov's scheme

$$\Phi_i^e = \frac{\Phi^e}{N_{\mathrm{dof}}^e} + \alpha \sum_{j \in \mathcal{N}_h^e} (u_i - u_j), \quad \alpha > 0$$

RD: Distributions process

Construction of a non-linear RD scheme

- First order monotone scheme, *e.g.* Rusanov's scheme

$$\Phi_i^e = \frac{\Phi^e}{N_{\mathrm{dof}}^e} + \alpha \sum_{j \in \mathcal{N}_h^e} (u_i - u_j), \quad \alpha > 0$$

- $\beta_i^e = \dfrac{\Phi_i^e}{\Phi^e}$ unbounded. Apply the limiting map $\beta_i^e \mapsto \hat{\beta}_i^e(u_h)$

$$\hat{\beta}_i^e(u^h) = \frac{\max(\beta_i^e, 0)}{\sum_{j \in \mathcal{N}_h^e} \max(\beta_j^e, 0)} \quad \Rightarrow \quad \hat{\beta}_i^e \in [0,1] \quad \& \sum_{i \in \mathcal{N}_h^e} \hat{\beta}_i^e = 1$$

RD: Distributions process

Construction of a non-linear RD scheme

- First order monotone scheme, *e.g.* Rusanov's scheme

$$\Phi_i^e = \frac{\Phi^e}{N_{\mathrm{dof}}^e} + \alpha \sum_{j \in \mathcal{N}_h^e} (u_i - u_j), \quad \alpha > 0$$

- $\beta_i^e = \dfrac{\Phi_i^e}{\Phi^e}$ unbounded. Apply the limiting map $\beta_i^e \mapsto \hat{\beta}_i^e(u_h)$

$$\hat{\beta}_i^e(u^h) = \frac{\max(\beta_i^e, 0)}{\displaystyle\sum_{j \in \mathcal{N}_h^e} \max(\beta_j^e, 0)} \quad \Rightarrow \quad \hat{\beta}_i^e \in [0,1] \quad \& \sum_{i \in \mathcal{N}_h^e} \hat{\beta}_i^e = 1$$

- Compute the high order distributed residual: $\hat{\Phi}_i^e = \hat{\beta}_i^e(u^h)\Phi^e$

RD: Distributions process

Construction of a non-linear RD scheme

- First order monotone scheme, *e.g.* Rusanov's scheme

$$\Phi_i^e = \frac{\Phi^e}{N_{\text{dof}}^e} + \alpha \sum_{j \in \mathcal{N}_h^e} (u_i - u_j), \quad \alpha > 0$$

- $\beta_i^e = \dfrac{\Phi_i^e}{\Phi^e}$ unbounded. Apply the limiting map $\beta_i^e \mapsto \hat{\beta}_i^e(u_h)$

$$\hat{\beta}_i^e(u^h) = \frac{\max(\beta_i^e, 0)}{\sum_{j \in \mathcal{N}_h^e} \max(\beta_j^e, 0)} \quad \Rightarrow \quad \hat{\beta}_i^e \in [0,1] \quad \& \sum_{i \in \mathcal{N}_h^e} \hat{\beta}_i^e = 1$$

- Compute the high order distributed residual: $\hat{\Phi}_i^e = \hat{\beta}_i^e(u^h)\Phi^e$
- Limiting enforces monotonicity but no upwinding included
- The solution consists in adding a filtering term

$$\hat{\Phi}_i^e = \hat{\beta}_i^e(u_h)\Phi^e + \epsilon_h \int_{\Omega_e} \Big(\boldsymbol{a} \cdot \boldsymbol{\nabla} \psi_i - \boldsymbol{\nabla} \cdot (\nu \boldsymbol{\nabla} \psi_i)\Big) \tau \left(\boldsymbol{a} \cdot \boldsymbol{\nabla} u_h - \boldsymbol{\nabla} \cdot (\nu \widetilde{\boldsymbol{\nabla} u_h})\right) \mathrm{d}\Omega$$

Something is still missing

Improvement of viscous term discretization. Inspired by H. Nishikawa

- Improve the discretization of diffusion dominated problems

Something is still missing

Improvement of viscous term discretization. Inspired by H. Nishikawa

- Improve the discretization of diffusion dominated problems
- Re-write the scalar equation as a first order system

$$\begin{cases} \nabla \cdot \boldsymbol{f}(u) - \nabla \cdot (\nu \boldsymbol{q}) = 0 \\ \boldsymbol{q} - \nabla u = 0 \end{cases}$$

Something is still missing

Improvement of viscous term discretization. Inspired by H. Nishikawa

- Improve the discretization of diffusion dominated problems
- Re-write the scalar equation as a first order system

$$\begin{cases} \nabla\cdot \boldsymbol{f}(u) - \nabla\cdot(\nu \boldsymbol{q}) = 0 \\ \boldsymbol{q} - \nabla u = 0 \end{cases}$$

- Discretize the FOS with a central scheme + a streamline stabilization

$$\int_{\Omega_e} \psi_i \begin{pmatrix} \nabla\cdot \boldsymbol{f}(u_h) - \nabla\cdot(\nu \boldsymbol{q}_h) \\ \boldsymbol{q}_h - \nabla u_h \end{pmatrix} + \int_{\Omega_e} \boldsymbol{A}\cdot\nabla\psi_i\, \boldsymbol{\tau} \begin{pmatrix} \nabla\cdot \boldsymbol{f}(u_h) - \nabla\cdot(\nu \boldsymbol{q}_h) \\ \boldsymbol{q}_h - \nabla u_h \end{pmatrix} = 0$$

Something is still missing

Improvement of viscous term discretization. Inspired by H. Nishikawa

- Improve the discretization of diffusion dominated problems
- Re-write the scalar equation as a first order system

$$\begin{cases} \nabla\cdot \boldsymbol{f}(u) - \nabla\cdot(\nu \boldsymbol{q}) = 0 \\ \boldsymbol{q} - \nabla u = 0 \end{cases}$$

- Discretize the FOS with a central scheme + a streamline stabilization

$$\int_{\Omega_e} \psi_i \begin{pmatrix} \nabla\cdot \boldsymbol{f}(u_h) - \nabla\cdot(\nu \boldsymbol{q}_h) \\ \boldsymbol{q}_h - \nabla u_h \end{pmatrix} + \int_{\Omega_e} \boldsymbol{A}\cdot\nabla\psi_i\, \boldsymbol{\tau} \begin{pmatrix} \nabla\cdot \boldsymbol{f}(u_h) - \nabla\cdot(\nu \boldsymbol{q}_h) \\ \boldsymbol{q}_h - \nabla u_h \end{pmatrix} = 0$$

- Replace $\boldsymbol{q}_h$ with $\widetilde{\nabla \mathrm{u}_h}$, the second line is discarded and the problem reads

$$\int_{\Omega_e} \psi_i \left(\nabla\cdot \boldsymbol{f}(u_h) - \nabla\cdot(\nu \widetilde{\nabla u_h})\right) + \int_{\Omega_e} \boldsymbol{a}\cdot\nabla\psi_i\, \tau_a \left(\boldsymbol{a}\cdot\nabla u_h - \nabla\cdot\left(\nu\widetilde{\nabla u_h}\right)\right)$$
$$+ \int_{\Omega_e} \nu\, \nabla\psi_i \cdot \left(\tau_d \left(\nabla u_h - \widetilde{\nabla u_h}\right)\right) = 0$$

Final form of the RD space discretization

Linear scheme

$$\Phi_i^{e,\mathrm{LW}} = \frac{\Phi^e}{N_{\mathrm{dof}}^e} + \int_{\Omega_e} \boldsymbol{a}\cdot\boldsymbol{\nabla}\psi_i\,\tau\left(\boldsymbol{a}\cdot\boldsymbol{\nabla}u_h - \boldsymbol{\nabla}\cdot\left(\nu\boldsymbol{\nabla}u_h\right)\right)\mathrm{d}\Omega + \int_{\Omega_e} \nu\,\boldsymbol{\nabla}\psi_i\cdot\left(\boldsymbol{\nabla}u_h - \widetilde{\boldsymbol{\nabla}u_h}\right)\mathrm{d}\Omega,$$

Non-linear scheme

$$\hat{\Phi}_i^{e,\mathrm{Rv}} = \hat{\beta}_i^{e,\mathrm{Rv}}(u_h)\,\Phi^e(u_h) + \epsilon_h^e(u_h)\int_{\Omega_e}\left(\boldsymbol{a}\cdot\boldsymbol{\nabla}\psi_i - \boldsymbol{\nabla}\cdot\left(\nu\boldsymbol{\nabla}\psi_i\right)\right)\tau\left(\boldsymbol{a}\cdot\boldsymbol{\nabla}u_h - \boldsymbol{\nabla}\cdot\left(\nu\boldsymbol{\nabla}u_h\right)\right)\mathrm{d}\Omega + \int_{\Omega_e}\nu\,\boldsymbol{\nabla}\psi_i\cdot\left(\boldsymbol{\nabla}u_h - \widetilde{\boldsymbol{\nabla}u_h}\right)\mathrm{d}\Omega,$$

The problem of the gradient reconstruction

- The "internal" gradient of the solution is replaced by a continuous approximation

$$\boldsymbol{\nabla}\mathsf{u}_h \quad \longrightarrow \quad \widetilde{\boldsymbol{\nabla}\mathsf{u}_h} = \sum_{i \in \mathcal{N}_h^e} \psi_i \widetilde{\boldsymbol{\nabla}\mathsf{u}_i}$$

- $\widetilde{\boldsymbol{\nabla}\mathsf{u}_i}$ is the reconstructed gradient of the solution at the DoF i

The problem of the gradient reconstruction

- The "internal" gradient of the solution is replaced by a continuous approximation

$$\nabla \mathsf{u}_h \quad \longrightarrow \quad \widetilde{\nabla \mathsf{u}_h} = \sum_{i \in \mathcal{N}_h^e} \psi_i \widetilde{\nabla \mathsf{u}_i}$$

- $\widetilde{\nabla \mathsf{u}_i}$ is the reconstructed gradient of the solution at the DoF i
- High-order preserving scheme
  - With $\mathsf{u}_h \in \mathbb{P}^k \Rightarrow (k+1)$-th accurate scheme for advection problems
  - How accurate the gradient reconstruction must be for advection-diffusion problems? As accurate as the solution

The problem of the gradient reconstruction

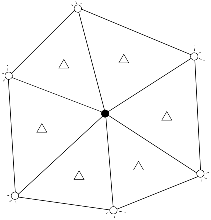
- The "internal" gradient of the solution is replaced by a continuous approximation

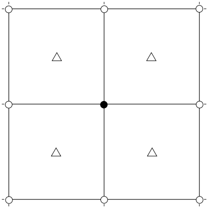
$$\nabla \mathsf{u}_h \quad \longrightarrow \quad \widetilde{\nabla \mathsf{u}_h} = \sum_{i \in \mathcal{N}_h^e} \psi_i \widetilde{\nabla \mathsf{u}_i}$$

- $\widetilde{\nabla \mathsf{u}_i}$ is the reconstructed gradient of the solution at the DoF i
- High-order preserving scheme
  - With $\mathsf{u}_h \in \mathbb{P}^k \Rightarrow (k+1)$-th accurate scheme for advection problems
  - How accurate the gradient reconstruction must be for advection-diffusion problems? As accurate as the solution
- Classical gradient reconstruction methods
  - Green-Gauss (area-weighted)
  - L^2 projection
  - Least-square
- With these approaches $\widetilde{\nabla \mathsf{u}_h} = \mathcal{O}(h^k)$. One order less than the solution

Super-convergent Patch Recovery (Zienkiewicz & Zhu, 92)
The main idea

- Idea coming from mechanical structure: compute stresses with the same accuracy of the displacements. Main idea: use it for CFD
- Gradients at certain points is more accurate than in others
- For structured grids theory identifies Gauss-Legendre points. No formal theory for unstructured grids
- Polynomial interpolation of degree k with a least square fitting to the sampled HO values within a patch of elements.





Super-convergent Patch Recovery method
How it works

- For each vertex i of the grid, the components of the of the reconstructed gradient are written in polynomial form

$$\left.\widetilde{\frac{\partial \mathsf{u}^h}{\partial x}}\right|_i = \boldsymbol{p}^{\mathrm{T}} \boldsymbol{a}_x, \quad \text{where} \quad \begin{aligned} \boldsymbol{p}^{\mathrm{T}} &= (1, x, y, x^2, \ldots, y^k) \\ \boldsymbol{a}_x &= (a_{x_1}, a_{x_2}, \ldots, a_{x_m}) \end{aligned}$$

Super-convergent Patch Recovery method
How it works

- For each vertex i of the grid, the components of the of the reconstructed gradient are written in polynomial form

$$\left.\widetilde{\frac{\partial \mathsf{u}^h}{\partial x}}\right|_i = \boldsymbol{p}^{\mathrm{T}} \boldsymbol{a}_x, \quad \text{where} \quad \begin{array}{l} \boldsymbol{p}^{\mathrm{T}} = (1, x, y, x^2, \dots, y^k) \\ \boldsymbol{a}_x = (a_{x_1}, a_{x_2}, \dots, a_{x_m}) \end{array}$$

- Minimize respect to $\boldsymbol{a}_x$ the function $F_x = \sum_{j=1}^{N_s} \left(\frac{\partial \mathsf{u}^h}{\partial x}(\boldsymbol{x}_j) - \boldsymbol{p}(\boldsymbol{x}_j)^{\mathrm{T}} \boldsymbol{a}_x \right)^2$

Super-convergent Patch Recovery method

How it works

- For each vertex i of the grid, the components of the of the reconstructed gradient are written in polynomial form

$$\left.\widetilde{\frac{\partial \mathsf{u}^h}{\partial x}}\right|_i = \boldsymbol{p}^{\mathrm{T}} \boldsymbol{a}_x, \quad \text{where} \quad \begin{array}{l} \boldsymbol{p}^{\mathrm{T}} = (1, x, y, x^2, \ldots, y^k) \\ \boldsymbol{a}_x = (a_{x_1}, a_{x_2}, \ldots, a_{x_m}) \end{array}$$

- Minimize respect to $\boldsymbol{a}_x$ the function $F_x = \sum_{j=1}^{N_s} \left(\frac{\partial \mathsf{u}^h}{\partial x}(\boldsymbol{x}_j) - \boldsymbol{p}(\boldsymbol{x}_j)^{\mathrm{T}} \boldsymbol{a}_x \right)^2$

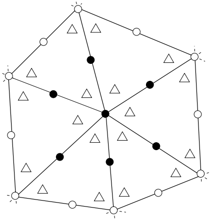
- Solve (in a least square sense) the linear system $A\boldsymbol{a}_x = \boldsymbol{b}_x^h$, for $\boldsymbol{a}_x$

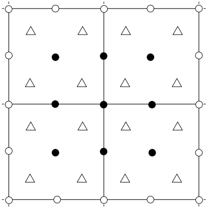
$$A = \begin{pmatrix} 1 & x_1 & y_1 & \ldots & y_1^k \\ 1 & x_2 & y_2 & \ldots & y_2^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N_s} & y_{N_s} & \ldots & y_{N_s}^k \end{pmatrix}, \qquad \boldsymbol{b}_x^h = \begin{pmatrix} \partial u^h/\partial x(\boldsymbol{x}_1) \\ \partial u^h/\partial x(\boldsymbol{x}_2) \\ \vdots \\ \partial u^h/\partial x(\boldsymbol{x}_{N_S}) \end{pmatrix}$$

ZZ Super-convergent Patch Recovery method

Important Remarks

- The method is very flexible: 2D/3D, hybrid grids
- $A \in \mathbb{R}^{N_S \times m}$, $N_s \geq m$ (N_s sampling points, m polynomial coefficients)
- On the boundary use the interior patch of the nearest vertices
- How to handle high order elements?



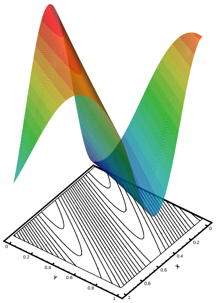


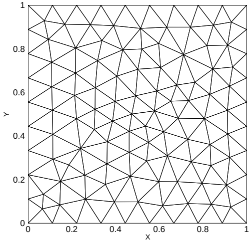
Gradient is reconstructed by evaluating on the patch, at the coordinates of the nodes, the polynomial function constructed for the nearest grid vertex

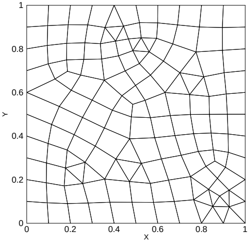
ZZ Super-convergent Patch Recovery method

Accuracy test

Gradient recovery of a smooth solution on unstructured grids

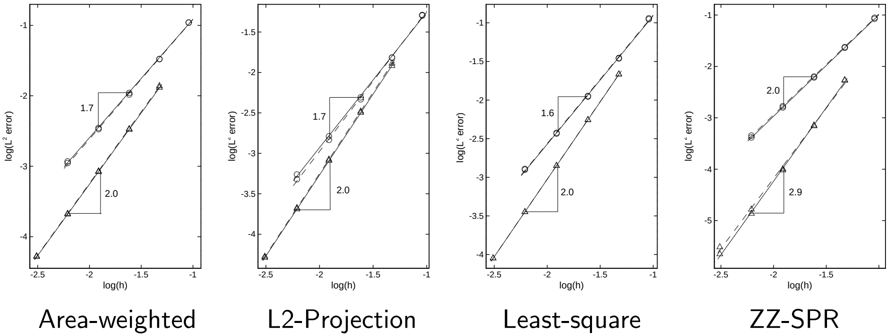






ZZ Super-convergent Patch Recovery method

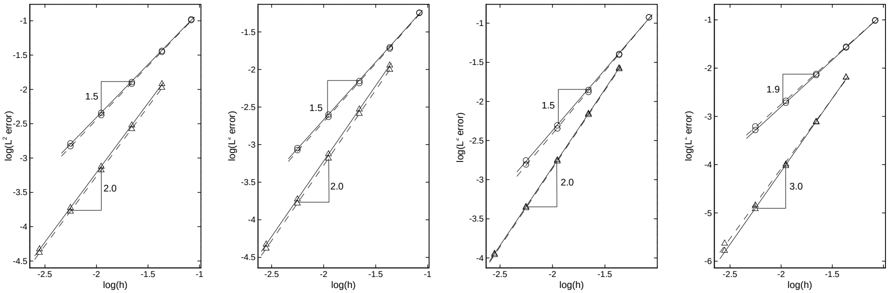
Accuracy test: triangular grids



ZZ-SPR gives gradients with the same order of accuracy of the solution

ZZ Super-convergent Patch Recovery method

Accuracy test: hybrid grids



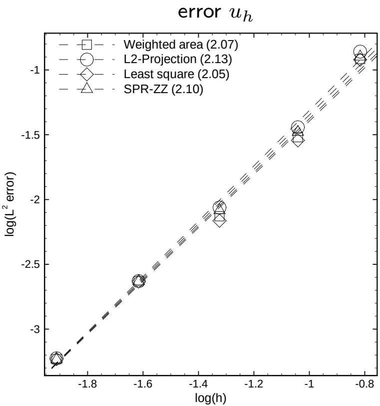
Area-weighted L2-Projection Least-square ZZ-SPR

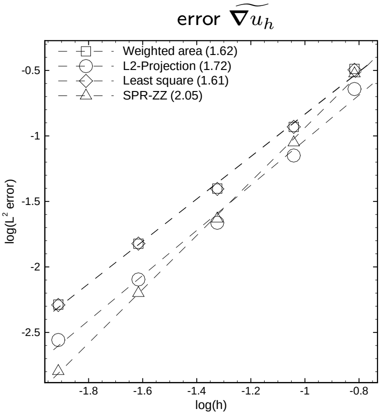
ZZ-SPR gives gradients with the same order of accuracy of the solution

SOLVE: $\boldsymbol{a} \cdot \boldsymbol{\nabla} u = \nu \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} u$, WITH $\nu = 0.01$ ($\mathrm{Pe} \sim 1$)

LINEAR SCHEME: EFFECT OF THE GRADIENT RECONSTRUCTION

- Triangular grids and $\mathbb{P}^1$ elements



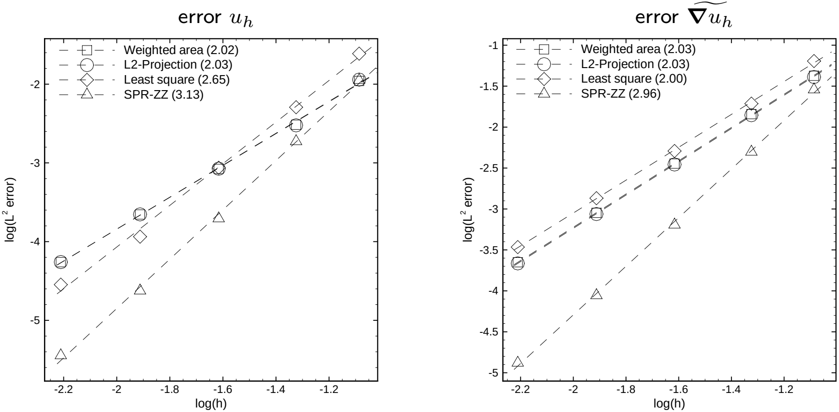


2nd order for solution and gradient with all gradient reconstruction

SOLVE: $\boldsymbol{a} \cdot \boldsymbol{\nabla} u = \nu \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} u$, WITH $\nu = 0.01$ (Pe ~ 1)

LINEAR SCHEME: EFFECT OF THE GRADIENT RECONSTRUCTION

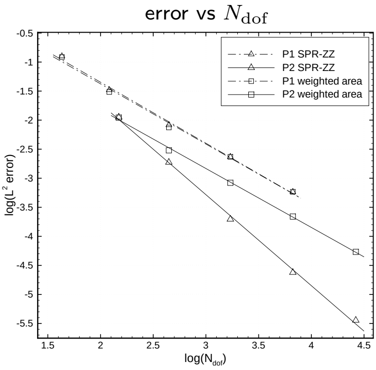
- Triangular grids and $\mathbb{P}^2$ elements

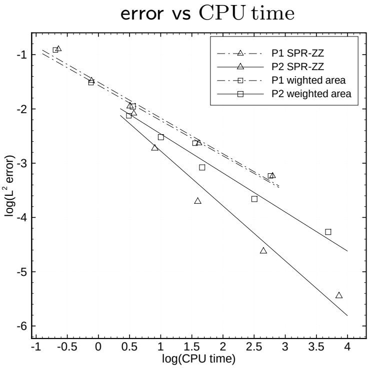


3rd order for solution and gradient only with ZPR-ZZ gradient reconstruction

SOLVE: $\boldsymbol{a} \cdot \boldsymbol{\nabla} u = \nu \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} u$, WITH $\nu = 0.01$ (Pe ~ 1)

LINEAR SCHEME: BENEFIT OF HIGH-ORDER APPROXIMATION

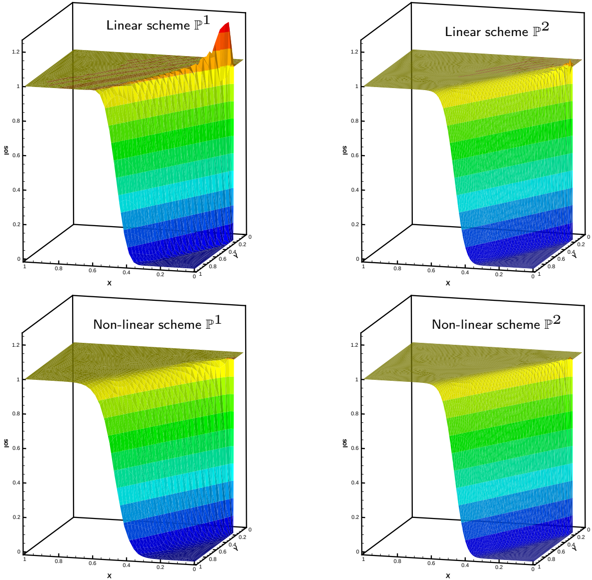




- For u^h error $\simeq 10^{-5}$
  - $\mathbb{P}^2: N_{\text{dof}} \simeq 12\,000,$ and CPU time $\simeq 25min$
  - $\mathbb{P}^1: N_{\text{dof}} \simeq 31\,000,$ and CPU time $\simeq 5h$

Discontinuous solution: linear and non-linear schemes

$\boldsymbol{a} \cdot \boldsymbol{\nabla} u = \nu \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} u$, $\boldsymbol{a} = (1/2, \sqrt{3}/2)^{\mathrm{T}}$, $\nu = 10^{-3}$



Governing equations
Compressible Navier-Stokes Equations

$$\boldsymbol{\nabla} \cdot \mathbf{f}^a(\mathsf{u}) - \boldsymbol{\nabla} \cdot \mathbf{f}^v(\mathsf{u}, \boldsymbol{\nabla}\mathsf{u}) = 0$$

$$\mathsf{u} = \begin{pmatrix} \rho \\ \boldsymbol{m} \\ E^t \end{pmatrix}, \quad \mathbf{f}^a = \begin{pmatrix} \boldsymbol{m} \\ \dfrac{\boldsymbol{m} \otimes \boldsymbol{m}}{\rho} + P\mathbb{I} \\ \left(E^t + P\right)\dfrac{\boldsymbol{m}}{\rho} \end{pmatrix}, \quad \mathbf{f}^v = \begin{pmatrix} 0 \\ \mathbb{S} \\ \mathbb{S} \cdot \dfrac{\boldsymbol{m}}{\rho} + \kappa \boldsymbol{\nabla} T \end{pmatrix}$$

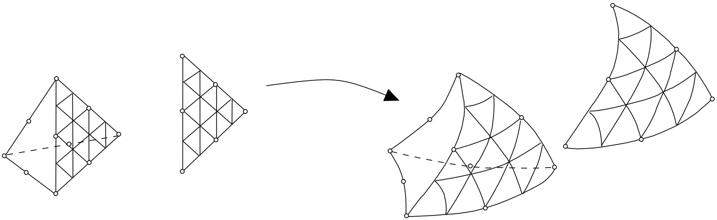
- Viscous flux function homogeneous with the respect to the gradient of the conservative variables: $\mathbf{f}^v(\mathsf{u}, \boldsymbol{\nabla}\mathsf{u}) = \mathbb{K}(\mathsf{u})\boldsymbol{\nabla}\mathsf{u}$
- Straightforward extension of the numerical schemes to system of equations
- Additional features respect to scalar equations
  - Boundary conditions
  - Implicit scheme

Boundary conditions
Boundary representation

- Imposition of the boundary conditions

$$\sum_{e \in \mathcal{E}_{h,i}} \Phi_i^e + \sum_{f \in \mathcal{F}_{h,i}} \Phi_i^{e,\partial} = 0, \quad \forall i \in \mathcal{N}_h,$$

- High-order schemes requires high-order boundary representation
- Here isoparametric formulation used. Same order for solution and geometry
- Piecewise polynomial approximation of the geometry

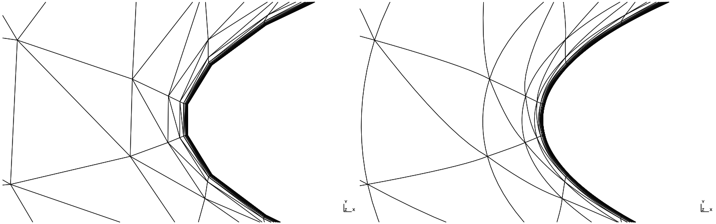


Boundary conditions
Boundary representation

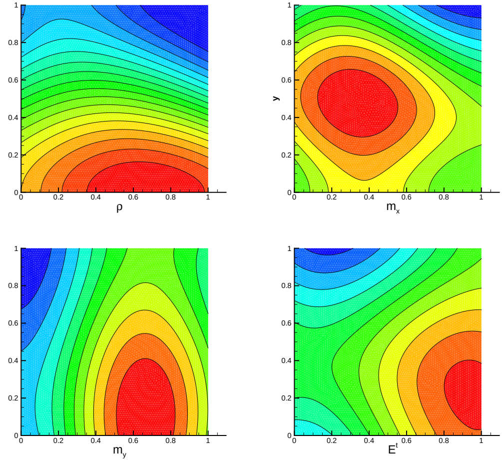
- Imposition of the boundary conditions

$$\sum_{e \in \mathcal{E}_{h,i}} \Phi_i^e + \sum_{f \in \mathcal{F}_{h,i}} \Phi_i^{e,\partial} = 0, \quad \forall i \in \mathcal{N}_h,$$

- High-order schemes requires high-order boundary representation
- Here isoparametric formulation used. Same order for solution and geometry
- Piecewise polynomial approximation of the geometry



Navier-Stokes manufactured solutions
Definition of the exact solution



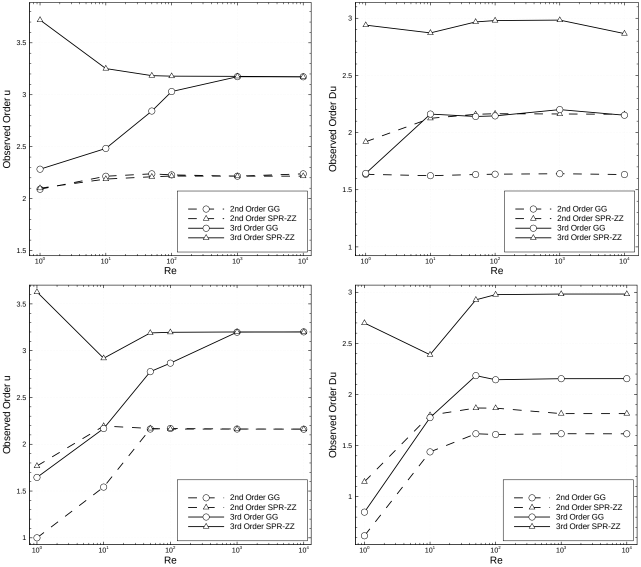
$$\boldsymbol{\nabla}\cdot\mathbf{f}^{a}(\mathsf{u}^{\mathrm{MS}})-\boldsymbol{\nabla}\cdot\mathbf{f}^{v}(\mathsf{u}^{\mathrm{MS}},\boldsymbol{\nabla}\mathsf{u}^{\mathrm{MS}})=\mathsf{S}(\mathsf{u}^{\mathrm{MS}})$$

Navier-Stokes manufactured solutions

Liner and Non-linear schemes: solution and gradient order of accuracy

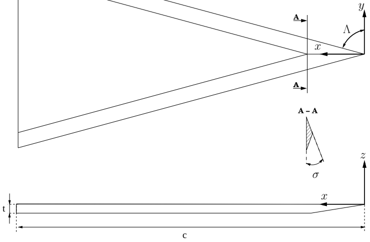
Linear Scheme

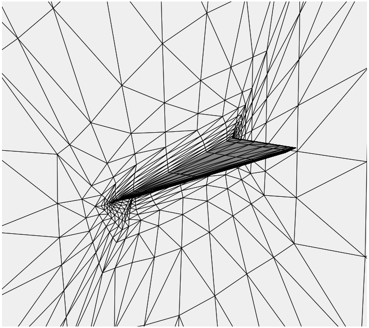
Non-linear Scheme



Laminar flow around a delta wing

$\mathrm{M} = 0.5$, $\alpha = 12.5°$, $\mathrm{Re} = 4\,000$

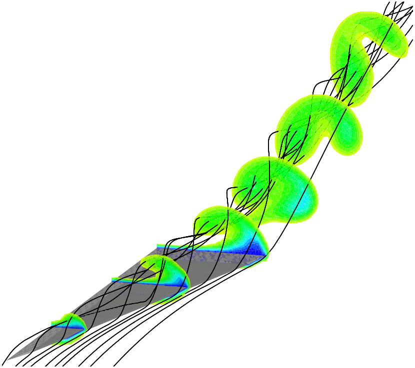




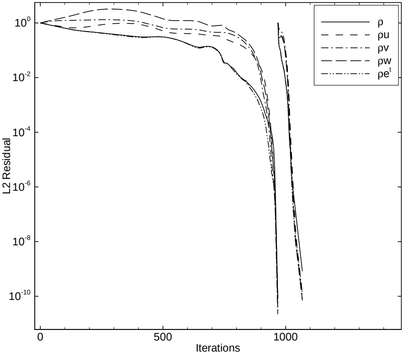
- Separated steady flow at high angle of attack
- Linear scheme with SPR-ZZ gradient reconstruction
- Three levels of nested grids. Parallel simulations: 8, 16, 32 processors
- Boundary conditions: no-slip adiabatic wall, symmetry plane and far-field
- Residual drop $\sim 10^{-10}$, respect to the initial value

Laminar flow around a delta wing

Example of solution and residual on the finest grid



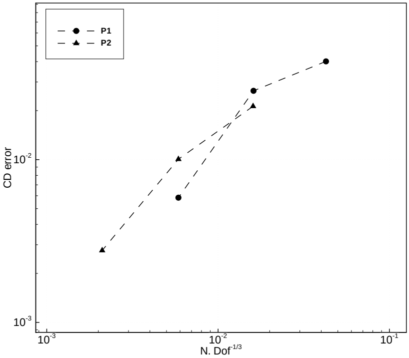
(a) Mach number and streamlines



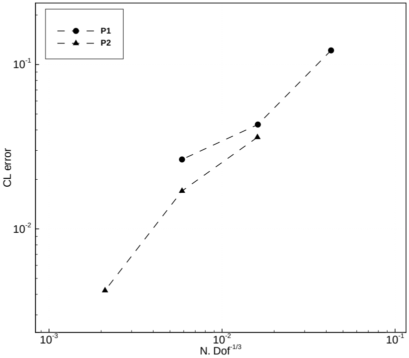
(b) Residual $\mathbb{P}^1$ and $\mathbb{P}^2$

Laminar flow around a delta wing

Force coefficients convergence



(a) Error CD

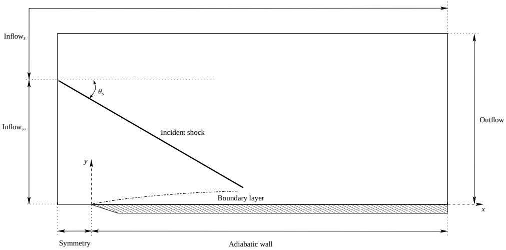


(b) Error CL

- Reference values: extrapolated from Higher-order DG resulted
- Effect on the CD of the singularity at the leading edge
- Benefit of high-order approximation on CL

Shock-Wave/Laminar Boundary Layer Interaction
Problem specifications

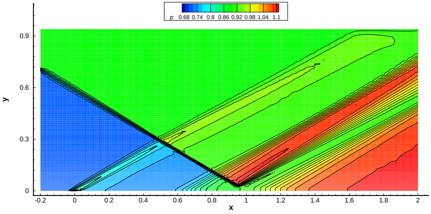
- $M_\infty = 2.15$, $\theta_s = 30.8°$, $\mathrm{Re}_{\mathrm{LS}} = 10^5$
- Non-linear scheme with SPR-ZZ gradient reconstruction
- Grid: $N_x = 90$ (uniform), $N_y = 85$ (clustered to the wall)
- Second and third order simulations
- Residual drop at least $\sim 10^{-8}$ with respect to the initial value



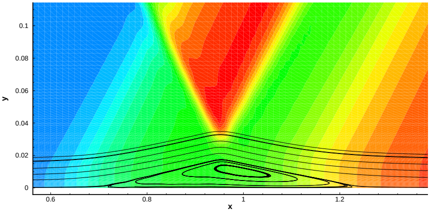
Shock-Wave/Laminar Boundary Layer Interaction

Problem specifications

- $M_\infty = 2.15$, $\theta_s = 30.8°$, $\mathrm{Re}_{\mathrm{LS}} = 10^5$
- Non-linear scheme with SPR-ZZ gradient reconstruction
- Grid: $N_x = 90$ (uniform), $N_y = 85$ (clustered to the wall)
- Second and third order simulations
- Residual drop at least $\sim 10^{-8}$ with respect to the initial value
- Example of third order solution



(a) Pressure contours

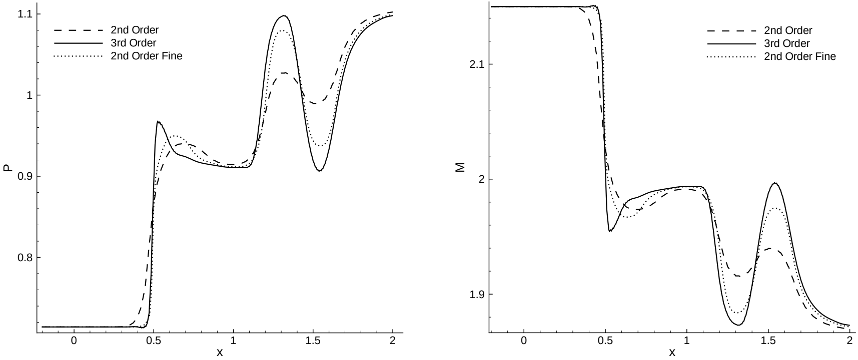


(b) Recirculation bubble

Shock-Wave/Laminar Boundary Layer Interaction

Comparison between second and third order accurate results

Profiles at $y = 0.29$



(a) Pressure (b) Mach number

Governing equations
RANS equations with Spalart-Allmaras model

$$\nabla \cdot \mathsf{f}^a(\mathsf{u}) - \nabla \cdot \mathsf{f}^v(\mathsf{u}, \nabla \mathsf{u}) = \mathsf{S}(\mathsf{u}, \nabla \mathsf{u})$$

$$\mathsf{u} = \begin{pmatrix} \rho \\ \boldsymbol{m} \\ E^t \\ \mu_t^\star \end{pmatrix}, \quad \mathsf{f}^a = \begin{pmatrix} \boldsymbol{m} \\ \dfrac{\boldsymbol{m} \otimes \boldsymbol{m}}{\rho} + P\mathbb{I} \\ \left(E^t + P\right)\dfrac{\boldsymbol{m}}{\rho} \\ \dfrac{\boldsymbol{m}}{\rho}\mu_t^\star \end{pmatrix}, \quad \mathsf{f}^v = \begin{pmatrix} 0 \\ \mathbb{S} \\ \mathbb{S} \cdot \dfrac{\boldsymbol{m}}{\rho} + \kappa \nabla T \\ \dfrac{\mu + \mu_t^\star}{\sigma} \nabla \left(\dfrac{\mu_t^\star}{\rho}\right) \end{pmatrix}$$

- $\mathsf{S} = (0, \mathbf{0}, 0, S_{\text{sa}})^{\text{T}}$, $S_{\text{sa}} = \mathcal{P}_{\text{sa}} + \mathcal{E}_{\text{sa}} - \mathcal{D}_{\text{sa}}$ Spalart-Allmaras source term
- Fully coupled approach: augmented advective and diffusive flux function
- Discretization with linear and non-linear schemes, similar to Navier-Stokes
- Additional work
  - Modifications to the original SA equation
  - Implicit solver

Spalart-Allmaras model
Improvement of the robustness

- Negative values of the turbulent working variable in the outer part of the boundary layer and wakes (insufficient mesh resolution)
- Clipping the eddy viscosity produce a physical valid model

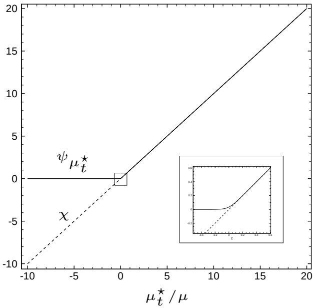
$$\mu_t = \begin{cases} \mu_t^\star f_{v1}, & \mu_t^\star > 0 \\ 0, & \mu_t^\star \leq 0 \end{cases}$$

  but robustness issues in the numerical solver are still present
- Several modification to the original SA model proposed
  - Not completely satisfying for HO methods

Spalart-Allmaras model
Improvement of the robustness (Peraire et al., 2011)

- A robust implementation of the SA equation requires to remove the adverse effects of negative values of $\mu_t^\star$ on the turbulence model
- $\mu_t^\star = \mu \dfrac{\mu_t^\star}{\mu} = \mu\chi$

$$\mu_t^\star = \mu\chi \quad \longrightarrow \quad \mu_t^\star = \mu\psi_{\mu_t^\star}$$



- The source term and flux functions of the SA equation tend to zero for $\mu_t^\star < 0$
- Differentiability of the equation

Implicit scheme for turbulent flows
Non-linear LU-SGS: Y.Sun, Z.J. Wang Y. Liu. 2009

- Jacobian-free approach non robust for HO RANS
- Solve with the symmetric variation of the Gauss-Seidel method with multiple sweeps ($k = 1, \ldots, k_{\max}$)

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathrm{R}_i}{\partial \mathrm{u}_i}\right] \Delta \mathrm{u}_i^{(k+1)} = -\mathrm{R}_i(\mathrm{u}^n) - \sum_{\substack{j \in \Omega_i \\ j \neq i}} \frac{\partial \mathrm{R}_i}{\partial \mathrm{u}_j} \Delta \mathrm{u}_j^{(*)}$$

$\Delta \mathrm{u}_j^{(*)}$ the most recently updated solution

- Linearization of the residual at $\mathrm{u}^{(*)} = \mathrm{u}^n + \Delta \mathrm{u}^{(*)}$

$$\begin{aligned} \mathrm{R}_i\left(\mathrm{u}^{(*)}\right) &\approx \mathrm{R}_i\left(\mathrm{u}^n\right) + \sum_{j \in \Omega_i} \frac{\partial \mathrm{R}_i}{\partial \mathrm{u}_j} \Delta \mathrm{u}_j^{(*)} \\ &= \mathrm{R}_i\left(\mathrm{u}^n\right) + \sum_{\substack{j \in \Omega_i \\ j \neq i}} \frac{\partial \mathrm{R}_i}{\partial \mathrm{u}_j} \Delta \mathrm{u}_j^{(*)} + \frac{\partial \mathrm{R}_i}{\partial \mathrm{u}_i} \Delta \mathrm{u}_i^{(*)} \end{aligned}$$

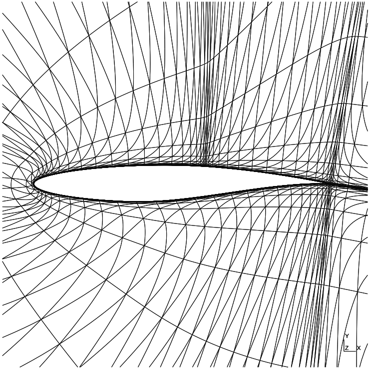
Implicit scheme for turbulent flows
Non-linear LU-SGS: Y.Sun, Z.J. Wang Y. Liu. 2009

- Substituting in the RHS of the SGS scheme

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}_i}{\partial \mathsf{u}_i}\right]\left(\Delta \mathsf{u}_i^{(k+1)} - \Delta \mathsf{u}_i^{(*)}\right) = -\mathsf{R}_i\left(\mathsf{u}^{(*)}\right) + \cancel{\frac{\Delta \mathsf{u}_i^{(*)}}{\Delta t^n}},$$

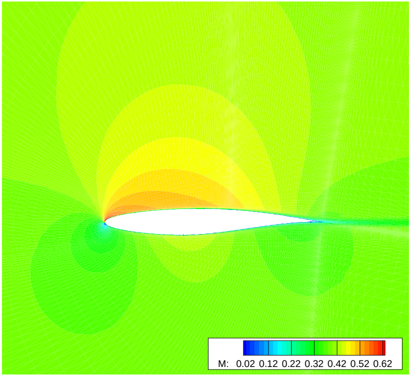
  which is solved with the forward and backward sweeps
- The RHS is nothing but the residual evaluated at the latest available solutions. (So the algorithm is called non-linear)
- At the beginning of each step the diagonal block of the Jacobian in the LHS is inverted using LU decomposition
- Only diagonal blocks of the approximated Jacobian used

Subsonic and transonic flow over a RAE2822 airfoil

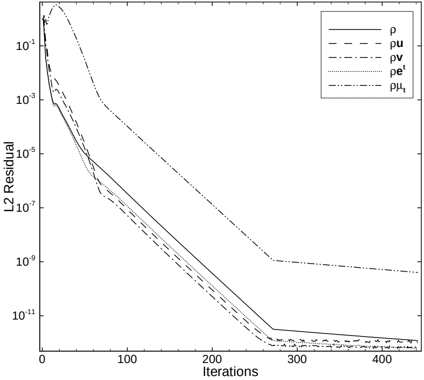


- $\mathrm{Re} = 6.5 \times 10^6$, $\alpha = 2.79°$
  - $\mathrm{M} = 0.4$ (subsonic)
  - $\mathrm{M} = 0.734$ (transonic)
- SPR-ZZ gradient reconstruction
  - Linear scheme (subsonic)
  - Non-linear scheme (transonic)
- Non-linear LU-SGS implicit scheme
  - Residual drop $\sim 10^{-10}$

Subsonic flow over a RAE2822 airfoil
Example of third order results on a fine grid



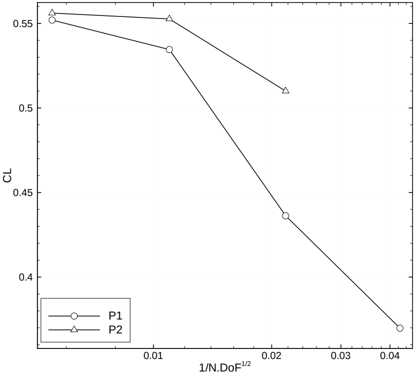
(a) Mach number contours



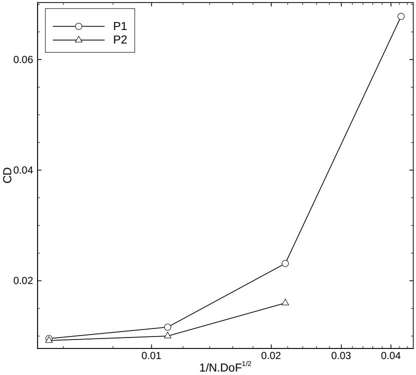
(b) Residual history

Subsonic flow over a RAE2822 airfoil

Force coefficient convergence



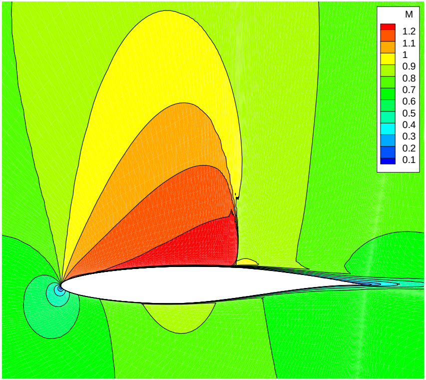
(a) Cl



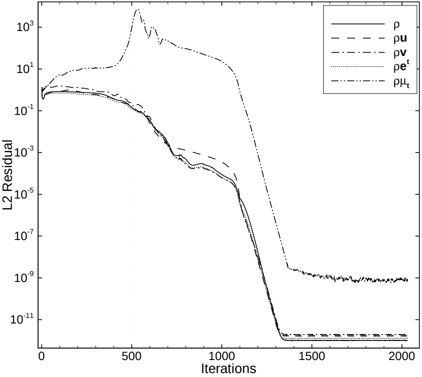
(b) Cd

Transonic flow over a RAE2822 airfoil

Example of third order results on a fine grid



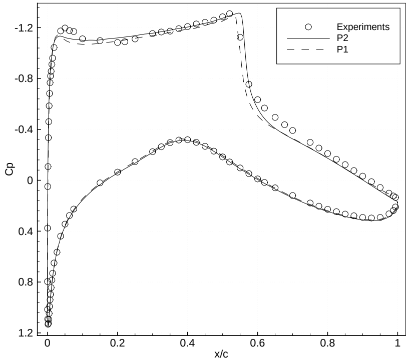
(a) Mach number contours

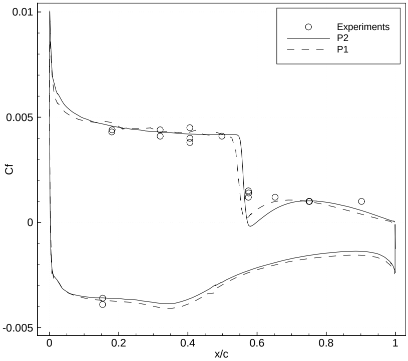


(b) Residual history

Transonic flow over a RAE2822 airfoil

Comparison second and third order results ($N_{\text{dof}} = 32\,784$)

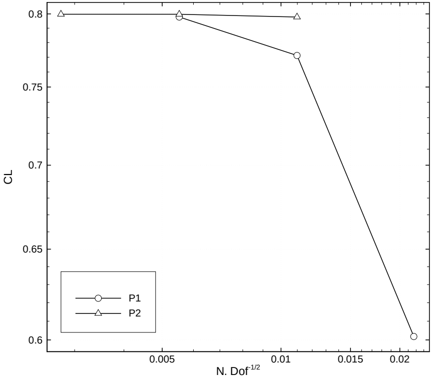




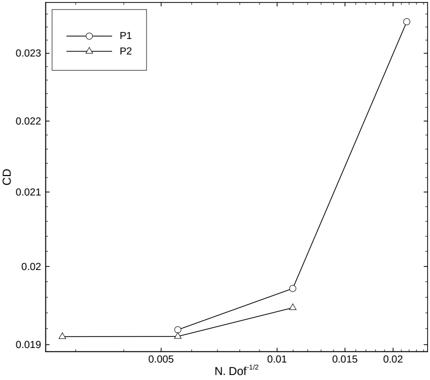
	Cl	Cd
RD 3rd ord.	0.7978	0.0192
RD 2nd ord.	0.7712	0.0197

	Cl	Cd
DG (UMich)	0.798	0.0191
Exp	0.803	0.0168

Transonic flow over a RAE2822 airfoil
Force coefficient convergence: DoFs



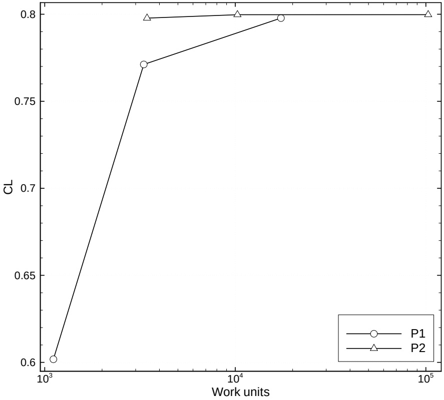
(a) Cl



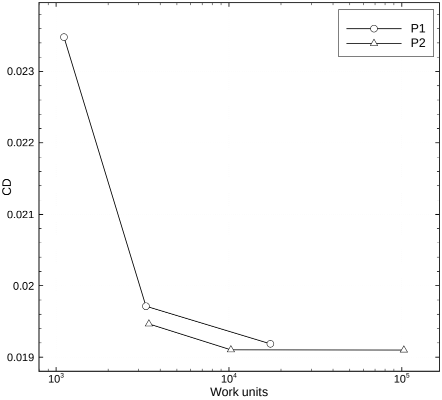
(b) Cd

Transonic flow over a RAE2822 airfoil

Force coefficient convergence: CPU time



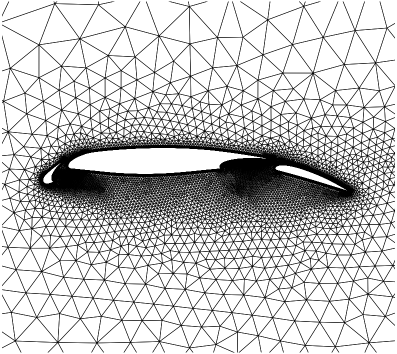
(a) Cl

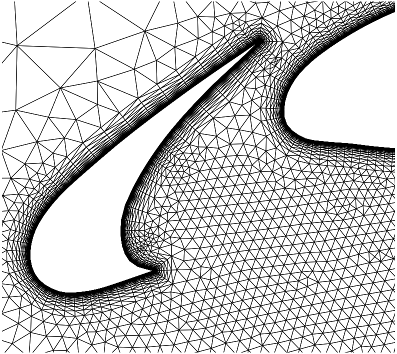


(b) Cd

L1T2 high-lift multi-element airfoil

$M = 0.197$, $Re = 3.52 \times 10^6$, $\alpha = 20.18°$

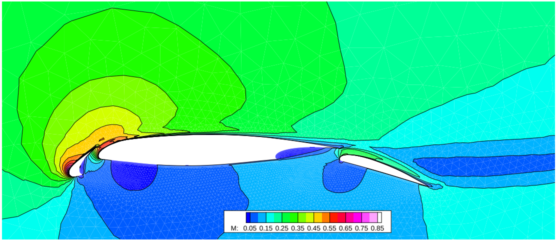


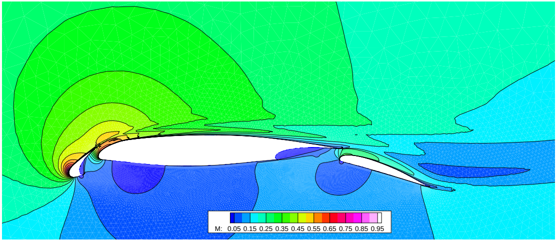


- Unstructured grid of triangles: $33\,338$ elements
- 2nd and 3rd order computations: linear scheme + SPR-ZZ
- Convergence criterion: normalized L2 residual $\sim 1 \times 10^{-10}$

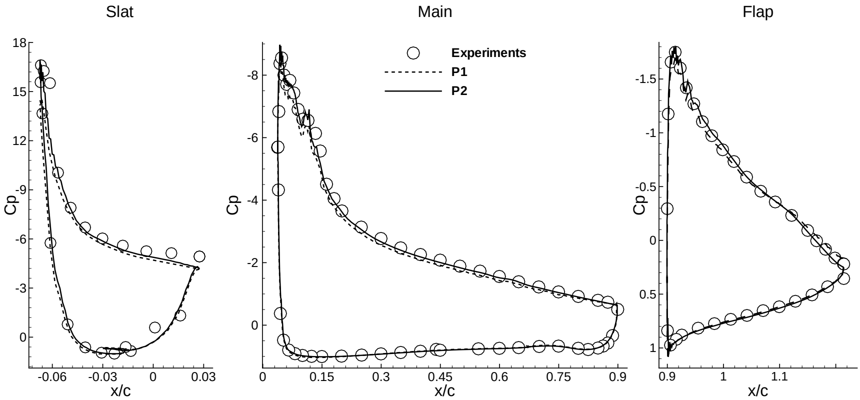
L1T2 high-lift multi-element airfoil

P1 & P2 Mach number contours





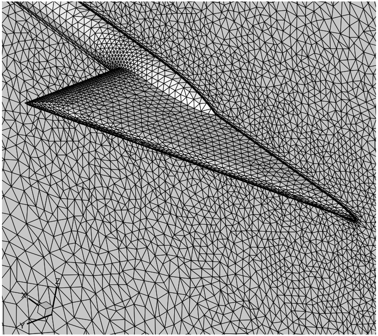
L1T2 high-lift multi-element airfoil
P1 & P2 Cp and experimental data



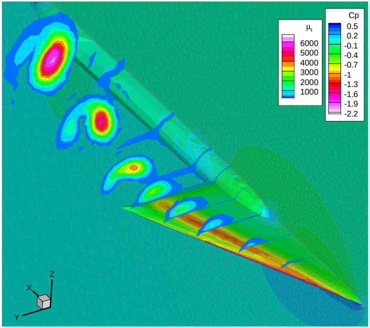
Subsonic flow over the NASA 65° sweep delta wing

$M = 0.4,\ \alpha = 13.3°,\ \mathrm{Re} = 3 \times 10^6$

- Linear scheme with SPR-ZZ
- Grid of $1\,145\,797$ tetrahedra
- Residual drop $\sim 10^{-5}$
- Parallel simulations: 96 processors



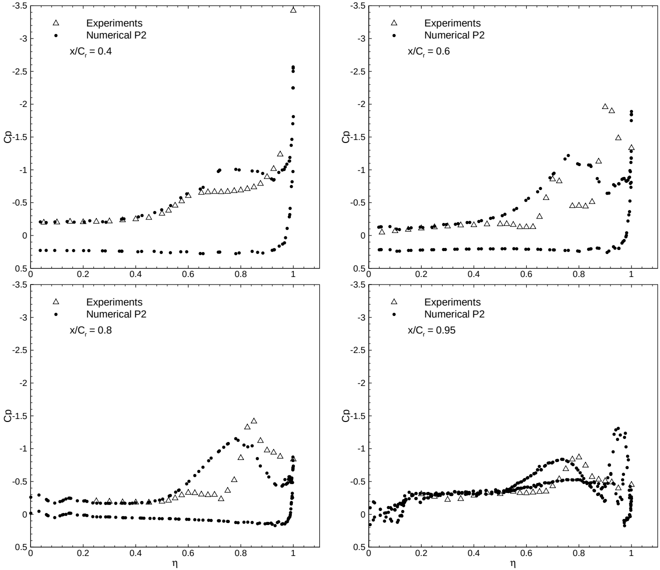
(a) Surface grid



(b) Cp and μ_t contours (third order)

Subsonic flow over the NASA 65° sweep delta wing

Third order results: Cp on the wing at different spanwise sections



Conclusions & Perspectives

Numerical method

- Idea developed first for advection-diffusion problems
- Based on accurate reconstruction of the gradient
- Possibility to get monotone & accurate solutions

Numerical results

- Extensive evaluation of the numerical solver
- Extension to N-S and RANS equations
- Efficient and robust implicit scheme
- Capabilities of HO approach in solving challenging applications
- Extension to complex EOS and hypersonic flows

Conclusions & Perspectives

Numerical method

- Idea developed first for advection-diffusion problems
- Based on accurate reconstruction of the gradient
- Possibility to get monotone & accurate solutions

Numerical results

- Extensive evaluation of the numerical solver
- Extension to N-S and RANS equations
- Efficient and robust implicit scheme
- Capabilities of HO approach in solving challenging applications
- Extension to complex EOS and hypersonic flows

Big challenge

- Unsteady problems: not clear how to get HO

Acknowledgment

- Prof. R. Abgrall (University of Zurich)
- Funded by the European FP7 STREP IDIHOM

Backup slides

- Write the original advection-diffusion problem as a first order system

$$\begin{cases} \boldsymbol{\nabla}\cdot \boldsymbol{f}(u) - \boldsymbol{\nabla}\cdot(\nu \boldsymbol{q}) = 0 \\ \boldsymbol{q} - \boldsymbol{\nabla} u = 0 \end{cases}$$

- Discretize the f.o.s with a central scheme + a streamline stabilization

$$\int_{\Omega_e} \psi_i \begin{pmatrix} \boldsymbol{\nabla}\cdot \boldsymbol{f}(u_h) - \boldsymbol{\nabla}\cdot(\nu \boldsymbol{q}_h) \\ \boldsymbol{q}_h - \boldsymbol{\nabla} u_h \end{pmatrix} + \int_{\Omega_e} \boldsymbol{A}\cdot\boldsymbol{\nabla}\psi_i \,\boldsymbol{\tau} \begin{pmatrix} \boldsymbol{\nabla}\cdot \boldsymbol{f}(u_h) - \boldsymbol{\nabla}\cdot(\nu \boldsymbol{q}_h) \\ \boldsymbol{q}_h - \boldsymbol{\nabla} u_h \end{pmatrix} = 0$$

where

$$\boldsymbol{A}\cdot\boldsymbol{\nabla}\psi_i = \begin{pmatrix} \boldsymbol{a}\cdot\boldsymbol{\nabla}\psi_i & -\nu\dfrac{\partial \psi_i}{\partial x} & -\nu\dfrac{\partial \psi_i}{\partial y} \\ -\dfrac{\partial \psi_i}{\partial x} & 0 & 0 \\ -\dfrac{\partial \psi_i}{\partial y} & 0 & 0 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\tau} = \begin{pmatrix} \tau_a & 0 & 0 \\ 0 & \tau_a & 0 \\ 0 & 0 & \tau_a \end{pmatrix}$$

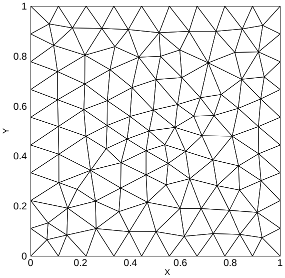
$$\frac{\partial u}{\partial t} + \boldsymbol{a} \cdot \boldsymbol{\nabla} u = \nu \left(\frac{\partial p}{\partial x} + \frac{\partial q}{\partial y} \right)$$
$$\frac{\partial p}{\partial t} = \frac{1}{T_r} \left(\frac{\partial u}{\partial x} - p \right)$$
$$\frac{\partial q}{\partial t} = \frac{1}{T_r} \left(\frac{\partial u}{\partial y} - q \right)$$

$$\frac{\partial \mathsf{u}}{\partial t} + \boldsymbol{A} \boldsymbol{\cdot} \boldsymbol{\nabla} \mathsf{u} = \mathsf{S},$$

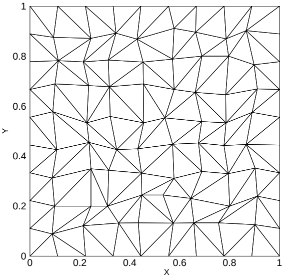
with

$$\mathsf{u} = \begin{pmatrix} u \\ p \\ q \end{pmatrix}, \quad A_x \begin{pmatrix} a_x & -\nu & 0 \\ \frac{1}{T_r} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_y \begin{pmatrix} a_y & 0 & -\nu \\ 0 & 0 & 0 \\ \frac{1}{T_r} & 0 & 0 \end{pmatrix}, \quad \boldsymbol{S} = \begin{pmatrix} 0 \\ -\frac{p}{T_r} \\ -\frac{q}{T_r} \end{pmatrix}$$

- Similar results with the non-linear scheme
- Similar results with grids of quadrangles and hybrid elements
- What is the effect of the grid regularity?

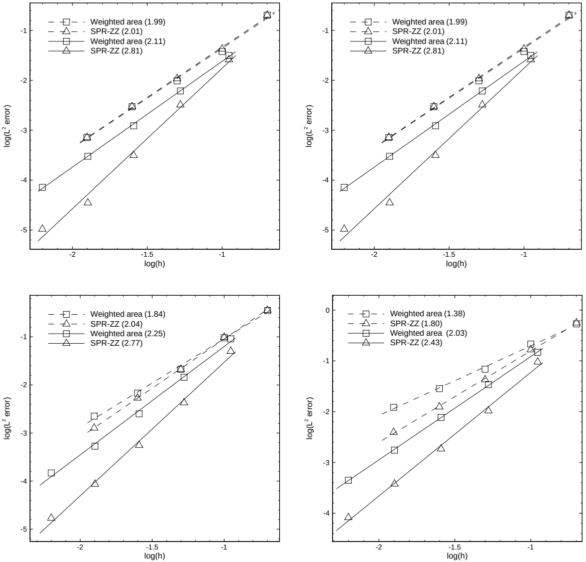


$\longrightarrow$



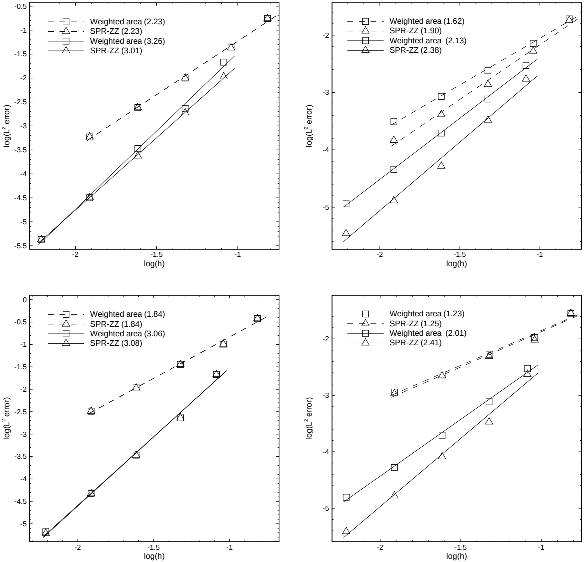
SOLVE: $\boldsymbol{a} \cdot \boldsymbol{\nabla} u = \nu \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} u$, WITH $\nu = 0.01$ (Pe ~ 1)

LINEAR SCHEME AND NON-LINEAR SCHEME ON PERTURBED GRID

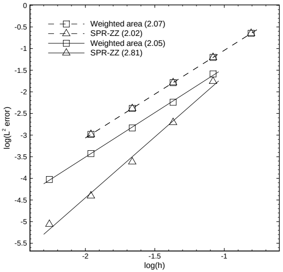


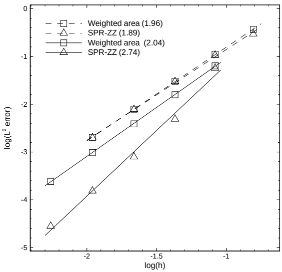
Solve: $\boldsymbol{a} \cdot \boldsymbol{\nabla} u = \nu \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} u$, with $\nu = 10^{-6}$ ($\mathrm{Pe} \gg 1$)

Linear scheme and non-linear scheme unstructured grid



- $-\boldsymbol{\nabla}\cdot\mathbb{K}\boldsymbol{\nabla}u^h = 0$
- $\mathbb{K} = \begin{pmatrix} 1 & 0 \\ 0 & \delta \end{pmatrix}, \delta = 10^3$





RD discretization of system of equations

- Calculation of the total residual

$$\Phi^e = \int_{\partial\Omega_e} \left(\mathbf{f}^a(\mathsf{u}_h) - \mathbb{K}(\mathsf{u}_h)\widetilde{\boldsymbol{\nabla}\mathsf{u}_h} \right) \cdot \hat{\boldsymbol{n}} \Big)$$

$\widetilde{\boldsymbol{\nabla}\mathsf{u}^h}$: gradient reconstruction similarly to scalar case

Linear scheme

$$\Phi_i^e = \frac{\Phi^e}{N_{\mathrm{dof}}^e} + \int_{\Omega_e} \boldsymbol{A} \cdot \boldsymbol{\nabla}\psi_i \, \tau \left(\boldsymbol{A} \cdot \boldsymbol{\nabla}\mathsf{u}_h - \boldsymbol{\nabla} \cdot \left(\mathbb{K}\widetilde{\boldsymbol{\nabla}\mathsf{u}_h} \right) \right)$$
$$+ \int_{\Omega_e} \mathbb{K} \cdot \boldsymbol{\nabla}\psi_i \left(\left(\boldsymbol{\nabla}\mathsf{u}_h - \widetilde{\boldsymbol{\nabla}\mathsf{u}_h} \right) \right)$$

$$\tau = \frac{|\Omega_e|}{N_{\mathrm{dim}}} \left(\sum_{i \in e} \mathsf{R}_{n_i} \Lambda_{n_i}^{+} \mathsf{L}_{n_i} + \mathbb{K}_{jj} \right)^{-1}$$

- Calculation of the total residual

$$\Phi^e = \int_{\partial\Omega_e} \left(\mathbf{f}^a(\mathsf{u}_h) - \mathbb{K}(\mathsf{u}_h)\widetilde{\nabla \mathsf{u}_h} \right) \cdot \hat{\boldsymbol{n}}$$

$\widetilde{\nabla \mathsf{u}^h}$: gradient reconstruction similarly to scalar case

Non-linear scheme

$$\hat{\Phi}_i^{e,\mathrm{Rv}} = \hat{\Phi}_i^e + \varepsilon_h^e(\mathsf{u}_h) \int_{\Omega_e} \left(\boldsymbol{A} \cdot \nabla \psi_i - \mathbb{K} \nabla \psi_i \right) \Xi \left(\boldsymbol{A} \cdot \nabla \mathsf{u}_h - \nabla \cdot \left(\mathbb{K} \widetilde{\nabla \mathsf{u}_h} \right) \right) \mathrm{d}\Omega$$
$$+ \int_{\Omega_e} \mathbb{K} \nabla \psi_i \cdot \left(\nabla \mathsf{u}_h - \widetilde{\nabla \mathsf{u}_h} \right) \mathrm{d}\Omega$$

$$\Xi = \frac{1}{2} |\Omega_e| \left(\sum_{i \in \mathcal{N}_h^e} \mathsf{R}_{n_i}(\bar{\mathsf{u}}) \Lambda_{n_i}^+(\bar{\mathsf{u}}) \mathsf{L}_{n_i}(\bar{\mathsf{u}}) + \sum_{j=1}^{N_{\mathrm{dim}}} \mathsf{K}_{jj}(\bar{\mathsf{u}}) \right)^{-1}.$$

- Imposition of weak boundary conditions

$$\sum_{e \in \mathcal{E}_{h,i}} \Phi_i^e + \sum_{f \in \mathcal{F}_{h,i}} \Phi_i^{e,\partial} = 0, \quad \forall i \in \mathcal{N}_h,$$

- Boundary residual contribution

$$\Phi_i^{e,\partial} = \int_{\partial\Omega_i \cap \partial\Omega} \psi_i \big(\mathbf{f}(\mathsf{u}^\partial) - \mathbf{f}(\mathsf{u}_h)\big) \cdot \boldsymbol{n} \ \mathrm{d}\partial\Omega$$

- Correction flux: $\big(\mathbf{f}(\mathsf{u}^\partial) - \mathbf{f}(\mathsf{u}_h)\big) \cdot \boldsymbol{n}$
  - Slip wall: $\Big(\mathbf{f}^a(\mathsf{u}^\partial_{\mathrm{wall}}) - \mathbf{f}^a(\mathsf{u}_h)\Big) \cdot \hat{\boldsymbol{n}} = -v_n(\rho,\, \rho\boldsymbol{v},\, E^t + P)^{\mathrm{T}}$
  - In/Out flow: $\Big(\mathbf{f}^a(\mathsf{u}^\partial_{\mathrm{in/out}}) - \mathbf{f}^a(\mathsf{u}^h)\Big) \cdot \hat{\boldsymbol{n}} = \mathsf{A}_n^-(\mathsf{u}^h)(\mathsf{u}^\partial_{\mathrm{in/out}} - \mathsf{u}^h)$
  - Adiabatic wall: $\boldsymbol{v} = 0$ (strong) and $\mathbf{f}^v(\mathsf{u}^\partial_{\mathrm{wall}}) = (0, 0, 0, -\kappa\widetilde{\boldsymbol{\nabla} T} \cdot \boldsymbol{n})^{\mathrm{T}}$

- Implicit Euler scheme with linearization: $A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n = -\mathsf{R}(\mathsf{u}_h^n)$

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right]\Delta\mathsf{u}^n = -\mathsf{R}(\mathsf{u}_h^n), \quad \Delta\mathsf{u}^n \equiv \mathsf{u}^{n+1} - \mathsf{u}^n$$

- Implicit Euler scheme with linearization: $A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n = -\mathsf{R}(\mathsf{u}_h^n)$

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right]\Delta\mathsf{u}^n = -\mathsf{R}(\mathsf{u}_h^n), \quad \Delta\mathsf{u}^n \equiv \mathsf{u}^{n+1} - \mathsf{u}^n$$

- Approximated solution $\|\mathsf{R}(\mathsf{u}_h^n) + A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n\| \leq \eta_h^n \|\mathsf{R}(\mathsf{u}_h^n)\|$ with GMRES

- Implicit Euler scheme with linearization: $A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n = -\mathsf{R}(\mathsf{u}_h^n)$

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right]\Delta\mathsf{u}^n = -\mathsf{R}(\mathsf{u}_h^n), \quad \Delta\mathsf{u}^n \equiv \mathsf{u}^{n+1} - \mathsf{u}^n$$

- Approximated solution $\|\mathsf{R}(\mathsf{u}_h^n) + A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n\| \leq \eta_h^n \|\mathsf{R}(\mathsf{u}_h^n)\|$ with GMRES
- Impossible to compute the analytical Jacobian: poor iterative convergence

- Implicit Euler scheme with linearization: $A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n = -\mathsf{R}(\mathsf{u}_h^n)$

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right]\Delta\mathsf{u}^n = -\mathsf{R}(\mathsf{u}_h^n), \quad \Delta\mathsf{u}^n \equiv \mathsf{u}^{n+1} - \mathsf{u}^n$$

- Approximated solution $\|\mathsf{R}(\mathsf{u}_h^n) + A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n\| \leq \eta_h^n \|\mathsf{R}(\mathsf{u}_h^n)\|$ with GMRES

JACOBIAN-FREE

$$A\mathsf{w} = \left(\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right)\mathsf{w}$$

- Implicit Euler scheme with linearization: $A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n = -\mathsf{R}(\mathsf{u}_h^n)$

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right]\Delta\mathsf{u}^n = -\mathsf{R}(\mathsf{u}_h^n), \quad \Delta\mathsf{u}^n \equiv \mathsf{u}^{n+1} - \mathsf{u}^n$$

- Approximated solution $\|\mathsf{R}(\mathsf{u}_h^n) + A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n\| \leq \eta_h^n \|\mathsf{R}(\mathsf{u}_h^n)\|$ with GMRES

Jacobian-free

$$A\mathsf{w} = \left(\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right)\mathsf{w}$$
$$\simeq \frac{\mathbb{I}}{\Delta t^n}\mathsf{w} + \frac{\mathsf{R}(\mathsf{u}_h^n + \epsilon\mathsf{w}) - \mathsf{R}(\mathsf{u}_h^n)}{\epsilon}$$

- Implicit Euler scheme with linearization: $A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n = -\mathsf{R}(\mathsf{u}_h^n)$

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right]\Delta\mathsf{u}^n = -\mathsf{R}(\mathsf{u}_h^n), \quad \Delta\mathsf{u}^n \equiv \mathsf{u}^{n+1} - \mathsf{u}^n$$

- Approximated solution $\|\mathsf{R}(\mathsf{u}_h^n) + A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n\| \leq \eta_h^n \|\mathsf{R}(\mathsf{u}_h^n)\|$ with GMRES

Jacobian-free

$$A\mathsf{w} = \left(\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right)\mathsf{w}$$

$$\simeq \frac{\mathbb{I}}{\Delta t^n}\mathsf{w} + \frac{\mathsf{R}(\mathsf{u}_h^n + \epsilon\mathsf{w}) - \mathsf{R}(\mathsf{u}_h^n)}{\epsilon}$$

$$\epsilon = \frac{\sqrt{1 + \|\mathsf{u}\|_{L_2}}}{\|\mathsf{w}\|_{L_2}}\epsilon_{\mathrm{rel}}, \quad \epsilon_{\mathrm{rel}} = 10^{-8}$$

- Implicit Euler scheme with linearization: $A(\mathsf{u}_h^n)\Delta \mathsf{u}_h^n = -\mathsf{R}(\mathsf{u}_h^n)$

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right]\Delta \mathsf{u}^n = -\mathsf{R}(\mathsf{u}_h^n), \quad \Delta \mathsf{u}^n \equiv \mathsf{u}^{n+1} - \mathsf{u}^n$$

- Approximated solution $\|\mathsf{R}(\mathsf{u}_h^n) + A(\mathsf{u}_h^n)\Delta \mathsf{u}_h^n\| \leq \eta_h^n \|\mathsf{R}(\mathsf{u}_h^n)\|$ with GMRES

JACOBIAN-FREE

$$A\mathsf{w} = \left(\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right)\mathsf{w}$$

$$\simeq \frac{\mathbb{I}}{\Delta t^n}\mathsf{w} + \frac{\mathsf{R}(\mathsf{u}_h^n + \epsilon \mathsf{w}) - \mathsf{R}(\mathsf{u}_h^n)}{\epsilon}$$

$$\epsilon = \frac{\sqrt{1 + \|\mathsf{u}\|_{L_2}}}{\|\mathsf{w}\|_{L_2}}\epsilon_{\mathrm{rel}}, \quad \epsilon_{\mathrm{rel}} = 10^{-8}$$

PRECONDITIONING: $AP^{-1}Px = b$

$$AP^{-1}\mathsf{w} = b \quad \text{and} \quad x = P^{-1}\mathsf{w}$$

- Implicit Euler scheme with linearization: $A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n = -\mathsf{R}(\mathsf{u}_h^n)$

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right]\Delta\mathsf{u}^n = -\mathsf{R}(\mathsf{u}_h^n), \quad \Delta\mathsf{u}^n \equiv \mathsf{u}^{n+1} - \mathsf{u}^n$$

- Approximated solution $\|\mathsf{R}(\mathsf{u}_h^n) + A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n\| \leq \eta_h^n \|\mathsf{R}(\mathsf{u}_h^n)\|$ with GMRES

Jacobian-free

$$A\mathsf{w} = \left(\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right)\mathsf{w}$$

$$\simeq \frac{\mathbb{I}}{\Delta t^n}\mathsf{w} + \frac{\mathsf{R}(\mathsf{u}_h^n + \epsilon\mathsf{w}) - \mathsf{R}(\mathsf{u}_h^n)}{\epsilon}$$

$$\epsilon = \frac{\sqrt{1 + \|\mathsf{u}\|_{L_2}}}{\|\mathsf{w}\|_{L_2}}\epsilon_{\mathrm{rel}}, \quad \epsilon_{\mathrm{rel}} = 10^{-8}$$

Preconditioning: $AP^{-1}Px = b$

$$AP^{-1}\mathsf{w} = b \quad \text{and} \quad x = P^{-1}\mathsf{w}$$

LU-SGS Preconditioner:

$$(D + L)D^{-1}(D + U)\,x = b + (LD^{-1}U)x$$

- Implicit Euler scheme with linearization: $A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n = -\mathsf{R}(\mathsf{u}_h^n)$

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right]\Delta\mathsf{u}^n = -\mathsf{R}(\mathsf{u}_h^n), \quad \Delta\mathsf{u}^n \equiv \mathsf{u}^{n+1} - \mathsf{u}^n$$

- Approximated solution $\|\mathsf{R}(\mathsf{u}_h^n) + A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n\| \leq \eta_h^n \|\mathsf{R}(\mathsf{u}_h^n)\|$ with GMRES

Jacobian-free

$$A\mathsf{w} = \left(\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right)\mathsf{w}$$

$$\simeq \frac{\mathbb{I}}{\Delta t^n}\mathsf{w} + \frac{\mathsf{R}(\mathsf{u}_h^n + \epsilon\mathsf{w}) - \mathsf{R}(\mathsf{u}_h^n)}{\epsilon}$$

$$\epsilon = \frac{\sqrt{1 + \|\mathsf{u}\|_{L_2}}}{\|\mathsf{w}\|_{L_2}}\epsilon_{\mathrm{rel}}, \quad \epsilon_{\mathrm{rel}} = 10^{-8}$$

Preconditioning: $AP^{-1}Px = b$

$$AP^{-1}\mathsf{w} = b \quad \text{and} \quad x = P^{-1}\mathsf{w}$$

LU-SGS Preconditioner:

$$(D + L)D^{-1}(D + U)\,x = b + \cancel{(LD^{-1}U)x}$$

- Implicit Euler scheme with linearization: $A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n = -\mathsf{R}(\mathsf{u}_h^n)$

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right]\Delta\mathsf{u}^n = -\mathsf{R}(\mathsf{u}_h^n), \quad \Delta\mathsf{u}^n \equiv \mathsf{u}^{n+1} - \mathsf{u}^n$$

- Approximated solution $\|\mathsf{R}(\mathsf{u}_h^n) + A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n\| \leq \eta_h^n \|\mathsf{R}(\mathsf{u}_h^n)\|$ with GMRES

Jacobian-free

$$A\mathsf{w} = \left(\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right)\mathsf{w}$$

$$\simeq \frac{\mathbb{I}}{\Delta t^n}\mathsf{w} + \frac{\mathsf{R}(\mathsf{u}_h^n + \epsilon\mathsf{w}) - \mathsf{R}(\mathsf{u}_h^n)}{\epsilon}$$

$$\epsilon = \frac{\sqrt{1 + \|\mathsf{u}\|_{L_2}}}{\|\mathsf{w}\|_{L_2}}\epsilon_{\mathrm{rel}}, \quad \epsilon_{\mathrm{rel}} = 10^{-8}$$

Preconditioning: $AP^{-1}Px = b$

$$AP^{-1}\mathsf{w} = b \quad \text{and} \quad x = P^{-1}\mathsf{w}$$

LU-SGS Preconditioner:

$$\underbrace{(D+L)D^{-1}(D+U)}_{P}x = b + \cancel{(LD^{-1}U)x}$$

- Implicit Euler scheme with linearization: $A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n = -\mathsf{R}(\mathsf{u}_h^n)$

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right]\Delta\mathsf{u}^n = -\mathsf{R}(\mathsf{u}_h^n), \quad \Delta\mathsf{u}^n \equiv \mathsf{u}^{n+1} - \mathsf{u}^n$$

- Approximated solution $\|\mathsf{R}(\mathsf{u}_h^n) + A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n\| \leq \eta_h^n \|\mathsf{R}(\mathsf{u}_h^n)\|$ with GMRES

Jacobian-free

$$A\mathsf{w} = \left(\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right)\mathsf{w}$$

$$\simeq \frac{\mathbb{I}}{\Delta t^n}\mathsf{w} + \frac{\mathsf{R}(\mathsf{u}_h^n + \epsilon\mathsf{w}) - \mathsf{R}(\mathsf{u}_h^n)}{\epsilon}$$

$$\epsilon = \frac{\sqrt{1 + \|\mathsf{u}\|_{L_2}}}{\|\mathsf{w}\|_{L_2}}\epsilon_{\text{rel}}, \quad \epsilon_{\text{rel}} = 10^{-8}$$

Preconditioning: $AP^{-1}Px = b$

$$AP^{-1}\mathsf{w} = b \quad \text{and} \quad x = P^{-1}\mathsf{w}$$

LU-SGS Preconditioner:

$$P = (D + L)D^{-1}(D + U)$$

- Implicit Euler scheme with linearization: $A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n = -\mathsf{R}(\mathsf{u}_h^n)$

$$\left[\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right]\Delta\mathsf{u}^n = -\mathsf{R}(\mathsf{u}_h^n), \qquad \Delta\mathsf{u}^n \equiv \mathsf{u}^{n+1} - \mathsf{u}^n$$

- Approximated solution $\|\mathsf{R}(\mathsf{u}_h^n) + A(\mathsf{u}_h^n)\Delta\mathsf{u}_h^n\| \leq \eta_h^n \|\mathsf{R}(\mathsf{u}_h^n)\|$ with GMRES

Jacobian-free

$$A\mathsf{w} = \left(\frac{\mathbb{I}}{\Delta t^n} + \frac{\partial \mathsf{R}}{\partial \mathsf{u}}(\mathsf{u}_h^n)\right)\mathsf{w}$$

$$\simeq \frac{\mathbb{I}}{\Delta t^n}\mathsf{w} + \frac{\mathsf{R}(\mathsf{u}_h^n + \epsilon\mathsf{w}) - \mathsf{R}(\mathsf{u}_h^n)}{\epsilon}$$

$$\epsilon = \frac{\sqrt{1 + \|\mathsf{u}\|_{L_2}}}{\|\mathsf{w}\|_{L_2}}\epsilon_{\mathrm{rel}}, \qquad \epsilon_{\mathrm{rel}} = 10^{-8}$$

Preconditioning: $AP^{-1}Px = b$

$$AP^{-1}\mathsf{w} = b \quad \text{and} \quad x = P^{-1}\mathsf{w}$$

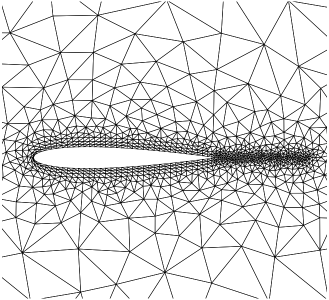
LU-SGS Preconditioner:

$$P = (D + L)D^{-1}(D + U)$$

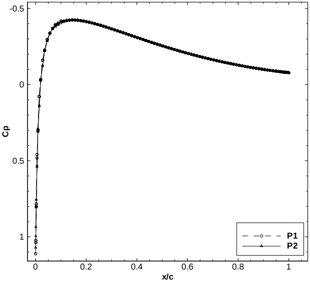
$$\begin{cases} x_i^\star = D_i^{-1}\left(\mathsf{w}_i - \sum_{j<i} \mathsf{w}_j\, x_j^\star\right), & i = 1, \ldots, N_{\mathrm{dof}} \\ x_i = x_i^\star - D_i^{-1} \sum_{j>i} \mathsf{w}_j\, x_j, & i = N_{\mathrm{dof}}, \ldots, 1 \end{cases}$$

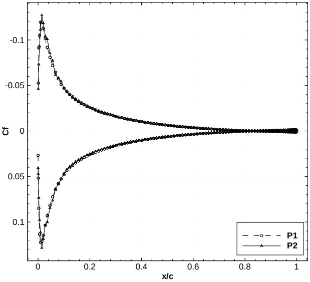
Laminar NACA-0012

$M = 0.5,\ \alpha = 0,\ \text{Re} = 5\,000$



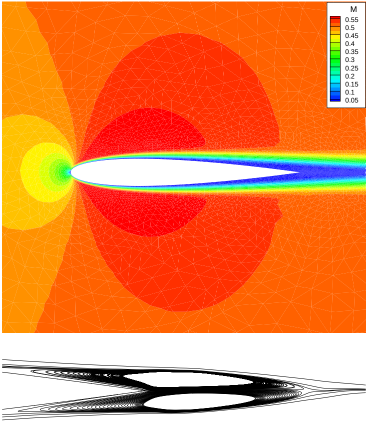
- 4216 P2 elements (8564 DOFs)
- Linear scheme with ZZ-SPR
- Residual down to zero machine



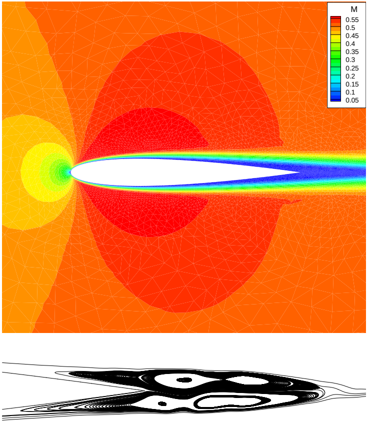


Laminar NACA-0012

$\mathrm{M} = 0.5,\ \alpha = 0,\ \mathrm{Re} = 5\,000$



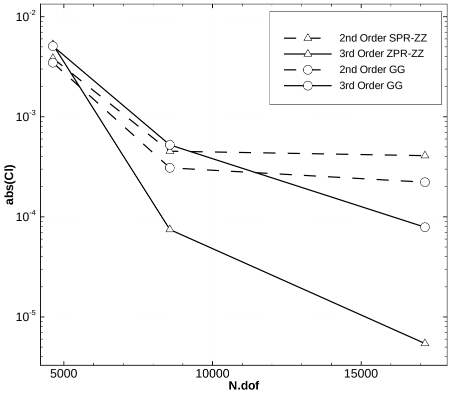
3rd order



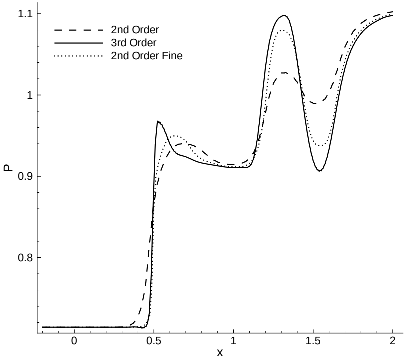
2nd order (Equivalent-DOF)

Laminar NACA-0012

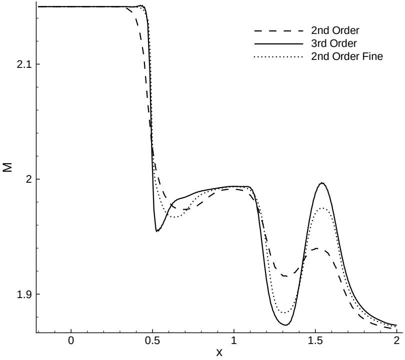
$M = 0.5,\ \alpha = 0,\ \mathrm{Re} = 5\,000$



Profiles at $y = 0.29$

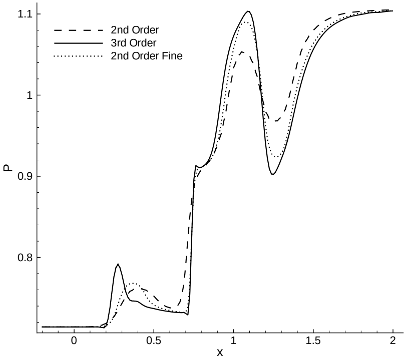


(a) Pressure

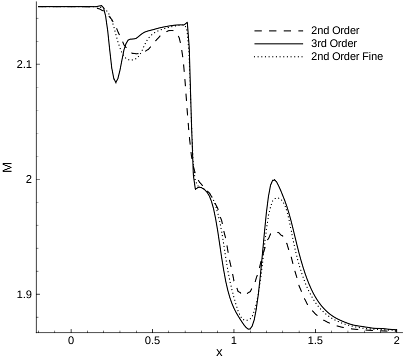


(b) Mach number

Profiles at $y = 0.15$

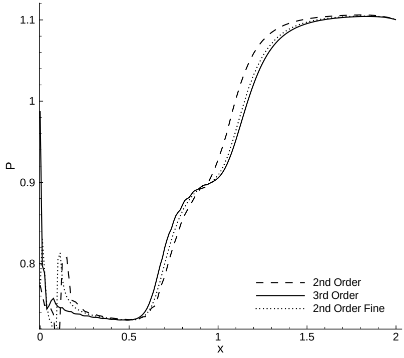


(a) Pressure

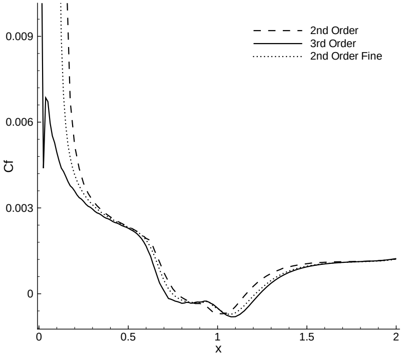


(b) Mach number

Profiles along the wall



(a) Pressure



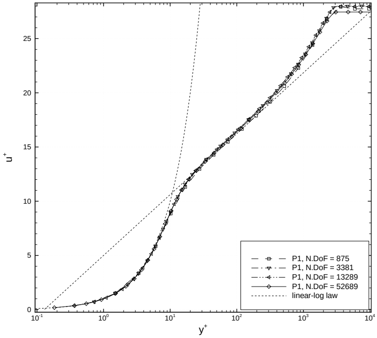
(b) Friction coefficient

Turbulent flow over a flat plate
$M = 0.2$, $\mathrm{Re}_{L=1} = 5 \times 10^6$

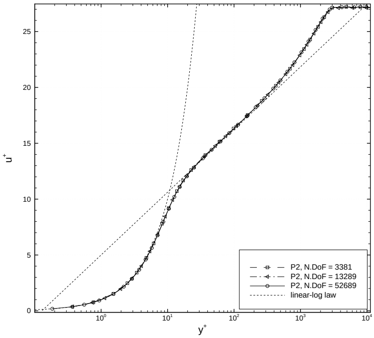
- Linear scheme with SPR-ZZ gradient reconstruction
- Jacobian-free with LU-SGS preconditioner (Residual drop $\sim 10^{-10}$)
- Nested grids
- Value of y_1^+ (at $x = 0.97$)

	Grid 35×25	Grid 69×49	Grid 137×97	Grid 273×193
$y^+_{1_{\mathrm{P1}}}$	1.500	0.722	0.359	0.182
$y^+_{1_{\mathrm{P2}}}$	0.765	0.372	0.184	–

- Linear scheme with SPR-ZZ gradient reconstruction
- Jacobian-free with LU-SGS preconditioner (Residual drop $\sim 10^{-10}$)
- Nested grids
- Velocity profiles (at $x = 0.97$)

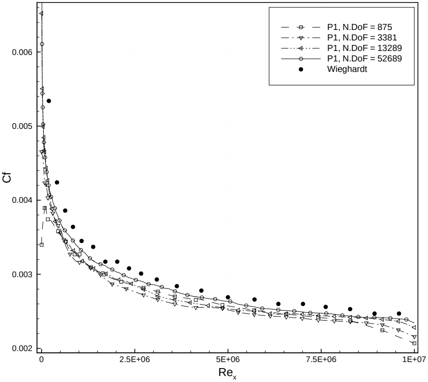


(a) Second order

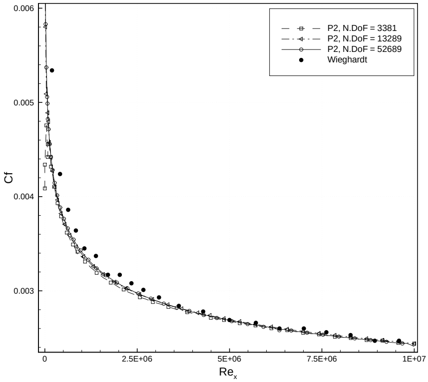


(b) Third order

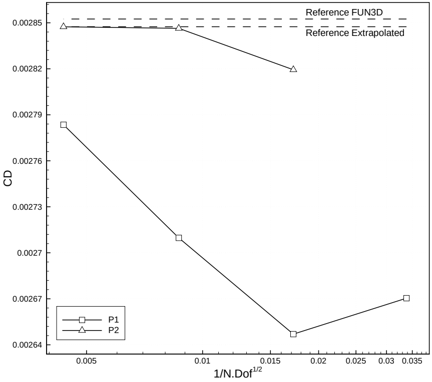
Friction coefficient along the plate



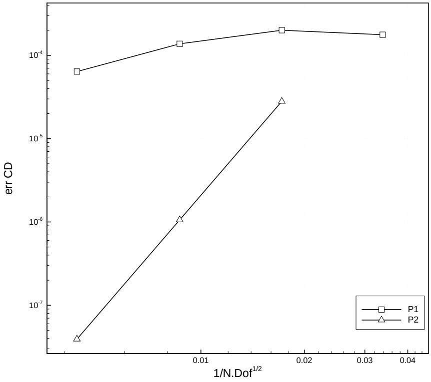
(a) Second order



(b) Third order



(a) Cd



(b) Cd error